CJ Jenks CPS 393

Question 1:

```
┌─────────────────────────────┐
│ Customers                   │
├─────────────────────────────┤
│ -customer_id (primary_key)  │
│ -name                       │
│ -address                    │
│ -phone                      │
│ -ssn                        │
└─────────────────────────────┘
```

```
┌─────────────────────────────┐
│ Account                     │
├─────────────────────────────┤
│ -account_id (primary_key)   │
│ -account_type               │
│ -balance                    │
└─────────────────────────────┘
```

```sql
CREATE TABLE Customers (
        customer_id INT AUTO_INCREMENT PRIMARY KEY,
        name VARCHAR(100),
        address VARCHAR(255),
        phone VARCHAR(15),
        ssn CHAR(9)
);

CREATE TABLE Accounts (
        account_id INT AUTO_INCREMENT PRIMARY KEY,
        account_type VARCHAR(20),
        balance VARCHAR(255),
);

CREATE TABLE CustomerAccount (
        customer_id INT,
        account_id INT,
        PRIMARY KEY (customer_id, account_id),
        FOREIGN KEY (customer_id) REFERENCES Customers(customer_id),
        FOREIGN KEY (account_id) REFERENCES Accounts(account_id),
);
```
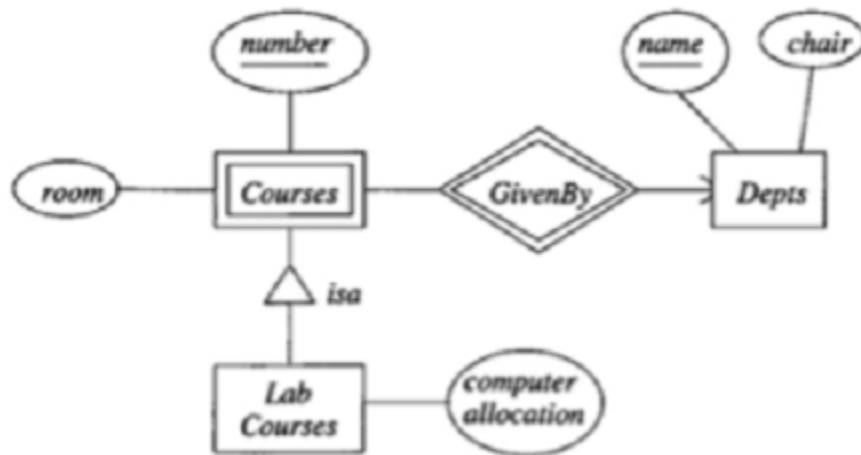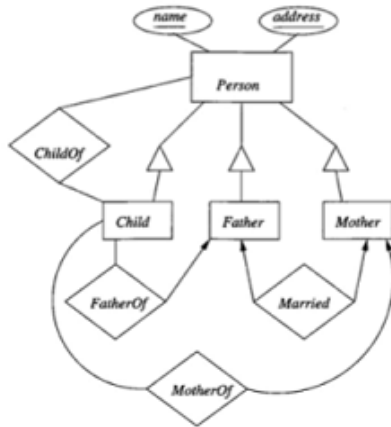
Question 2:

Figure 4.32: E/R diagram for Exercise 4.6.1

```
CREATE TABLE Departments (
        name VARCHAR(50) PRIMARY KEY,
        chair VARCHAR(100)
);

CREATE TABLE Courses (
        number INT PRIMARY KEY,
        room VARCHAR(50),
        dept_name VARCHAR(50),
        FOREIGN KEY (dept_name) REFERENCES Departments(name)
);

CREATE TABLE LabCourses (
        number INT PRIMARY KEY,
        computer_allocation VARCHAR(50),
        FOREIGN KEY (number) REFERENCES Courses(name)
);
```

```
CREATE TABLE Person (
        person_id INT AUTO_INCREMENT PRIMARY KEY,
        name VARCHAR(100),
        address VARCHAR(255)
);

CREATE TABLE ChildOf (
    child_id INT,
    father_id INT,
    mother_id INT,
    PRIMARY KEY (child_id),
    FOREIGN KEY (child_id) REFERENCES Person(person_id),
    FOREIGN KEY (father_id) REFERENCES Person(person_id),
    FOREIGN KEY (mother_id) REFERENCES Person(person_id)
);

CREATE TABLE Married (
    father_id INT,
    mother_id INT,
    PRIMARY KEY (father_id, mother_id),
    FOREIGN KEY (father_id) REFERENCES Person(person_id),
    FOREIGN KEY (mother_id) REFERENCES Person(person_id)
);
```

Question 3:
SHOW TABLES;

SELECT COUNT(*) AS num_of_cities FROM city WHERE CountryCode = 'USA';

SELECT COUNT(*) AS num_of_cities_starting_with_a FROM city WHERE Name LIKE 'A%';

```sql
SELECT POPULATION, AVG(LifeExpectancy) AS avg_life_expectancy FROM country WHERE
Code LIKE 'ARG';

SELECT Name, LifeExpectancy FROM country ORDER BY LifeExpectancy DESC LIMIT 1;

SELECT Name, FROM city WHERE Name LIKE 'F%' LIMIT 5;
```